

Lasse Häll

TUTORIAALIN SUUNNITTELU JA TOTEUTTAMINEN MMO-PELIIN

TUTORIAALIN SUUNNITTELU JA TOTEUTTAMINEN MMO-PELIIN

Lasse Häll
Opinnäytetyö
Syksy 2015
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, Web-sovelluskehitys

Tekijä(t): Lasse Häll

Opinnäytetyön nimi: Tutoriaalin suunnittelu ja toteuttaminen MMO-peliin

Työn ohjaaja: Matti Viitala

Työn valmistumislukukausi- ja vuosi: Syksy 2015

Sivumäärä: 24

Työn päämääränä oli suunnitella ja toteuttaa ensimmäiset versiot Warcaster-pelin tutoriaalista paikalliselle startup-yritys Helmeni Oy:lle.

Kehitystyö tapahtui ketteriä menetelmiä hyödyntäen pääasiassa päivittäisillä stand-up tapaamisilla viikon mittaisissa sprinteissä. Suunnittelu ja kehitys tehtiin iteratiivisesti ja todella nopealla tahdilla.

Työn tavoitteena oli luoda pelin tarkoitusperiä palveleva tutoriaali-osuus, jolla on suuri merkitys pelaajien kiinnostuksen herättämisessä ja ylläpitämisessä itse peliin. Nopeasta iteratiivisesta kehitystyylistä johtuen tutoriaalin suunnittelu pohjautui lähinnä muihin vastaavan kaltaisiin peleihin ja ajankohtaiseen pelisuunnittelun kirjallisuuteen.

Työn päättyessä itse peli oli edelleen suurimmalta osin kehitysvaiheessa, joten pelin lopullista versiota varten tutoriaalia ei käytännössä pystynyt suunnittelemaan ja toteuttamaan. Valmiiksi saatiin pelin alkuvaiheen tutoriaali-tehtävät, jotka opettavat miten peliä pelataan. Lisäksi työn aikana toteutettiin käyttöliittymän esittelyruutu.

Asiasanat: pelisuunnittelu, pelikehitys, unity, ui, ux

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Web-application development

Author(s): Lasse Häll
Title of thesis: Designing and implementing a tutorial into an MMO-game
Supervisor(s): Matti Viitala
Term and year when the thesis was submitted: Autumn 2015 Number of pages: 24

The aim of the thesis was to design and implement the first versions of tutorial in a game called Warcaster for a local startup company Helmeni Oy.

Development was done by agile software development methods mainly through daily stand-up meetings in week-long sprints. Design and development was done iteratively and in a very quick pace.

The aim of my thesis was to create a functional tutorial that plays a big part in creating and keeping interest of the players. Due to the quick iterative process, the design was based largely on similar game titles and current game design literature.

At the end of the work period the game itself was still mostly in development phase, so designing and implementing of the tutorial for the final version of the game wasn't possible. The tutorial missions that show how to play the game in the beginning of the game were completed. Additionally, an introduction screen for the user interface was implemented.

Keywords: game design, game development, unity, ui, ux

SISÄLLYS

1	JOHDANTO	6
2	WARCASTER.....	7
3	SUUNNITTELU.....	8
3.1	Tutoriaali.....	8
3.2	Graafinen käyttöliittymä	10
3.3	Käyttäjäkokemus	12
4	TOTEUTTAMINEN	13
4.1	Unity	13
4.1.1	Editori.....	13
4.1.2	MonoDevelop IDE.....	15
4.2	Git.....	17
4.3	Ketterät menetelmät	17
5	GAMESCOM	20
6	POHDINTA	21
	LÄHTEET.....	22
	LIITTEET	23

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli tuottaa ensimmäiset versiot laadukkaasta ja tarkoituksperiään hyvin palvelevasta tutoriaali-osuudesta Warcaster-nimiseen MMORTS-genren mobiilipeliin. Opinnäytetyössä tarkastellaan tutoriaalin käsitettä ja pyritään ymmärtämään millainen on hyvin suunniteltu ja toteutettu mobiilipelin tutoriaali. Lisäksi työssä esitellään yleisiä pelikehityksessä käyttämiämme työkaluja ja menetelmiä. Kehitin yrityksen työntekijöiden kanssa yhteistyössä tutoriaalin ohella myös muuta peliin liittyvää suunnittelua ja sisältöä pääasiallisesti Unity-pelimootoria käytäen.

Työni aiheen löysin Oulu Game Labissa suorittamani harjoittelupaikan kautta tuntemastani startup-yrityksestä. Kyseessä oli yrityksen ensimmäinen pelituote, joka työni aloituksen aikana oli kehityksen alkuvaiheessa. Aihe oli kiinnostava ja pitää sisällään paljon oleellista pelisuunnittelun teoriaa ja pelikehityksen käytäntöä.

Tutoriaali on pelin menestyksen kannalta tärkeä, sillä se vaikuttaa oleellisesti siihen, miten houkuttelevan ja mielenkiintoisen ensivaikutelman peli luo itse pelaajille. Tutoriaalin tarkoituksena on opettaa pelaajalle, miten peliä pelataan ja mistä pelissä on kaiken kaikkiaan kysymys. Jos tämä osuus ei tunnu pelaajasta vaivattomalta, on pelaajan kiinnostus todennäköistä menettää jo ensi-minuuteilla.

2 WARCASTER

Warcaster on genreltään MMORTS eli Massively Multiplayer Online Realtime Strategy Game (Massiivinen moninpelattava reaali-aika strategia) ja sitä voi pelata sekä selaimella että älypuhelimilla ja tableteilla. Muita tunnettuja MMORTS pelejä ovat esimerkiksi Travian, Clash of Clans ja Age of Empires Online, joista selainpeli Travian muistuttaa eniten Warcasteria pelityyliltään.

Warcaster on Reaaliaikainen massiivimultipelaaja strategiapeli, jossa pelaaja haastetaan uudeksi fantasiamaailman hallitsijaksi. Peli alkaa post-apokalyptisesta tilanteesta, jossa on juuri päättynyt massiivinen magiantäyteinen sota. Kaikki pelaajat aloittavat rähjäisestä kylästä omaksuen magiaa hallitsevan Warcasterin hahmon. Pelaajan tärkein päätehtävä alussa on kasvattaa kylänsä taloudellisesti ja sotilaallisesti johtavaan tilanteeseen ympärillä oleviin kilpailijoihinsa nähden.

Warcaster harjaantuu pelin myötä voimakkaammaksi velhoksi. Kasvavan valtansa ja johtajuutensa avulla Warcaster pyrkii levittämään kylänsä vaikutusvaltaa laajemmalle myös muiden Warcasterien kanssa yhteistyössä. Kommunikaatio ja yhteistyö ovat tärkeässä strategisessa asemassa pelaajan kannalta alusta alkaen.

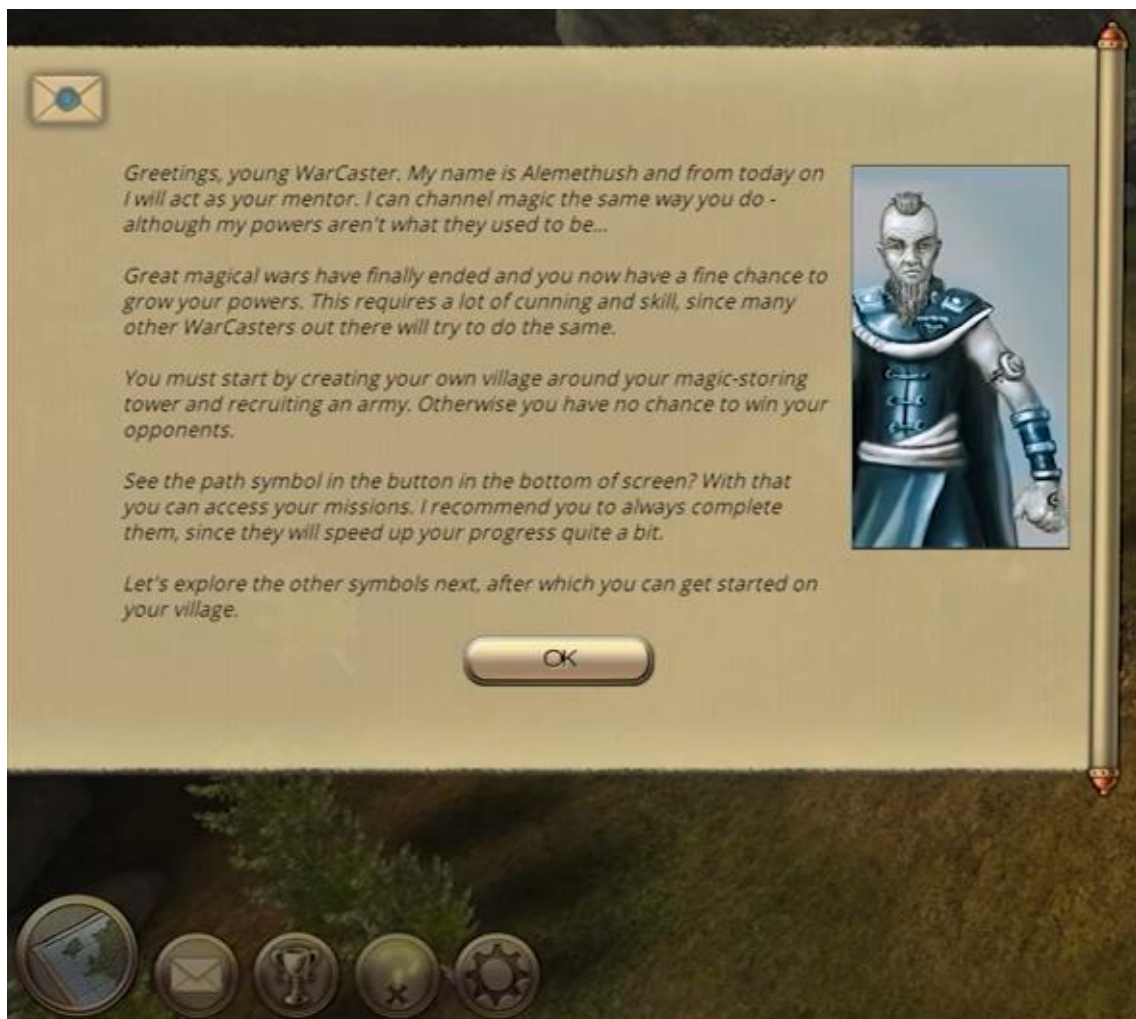
Pelin elinkaareen kuuluu alkuvaiheen kasvu, keskivaiheen sotiminen ja loppuvaiheessa erilaisten pitkäkestoisten ja vaativien tehtävien suorittaminen. Tehtävät ovat tarinapohjaisia ja antavat pelaajille suuntauksen toimilleen. Loppuvaiheen tehtävien suorittaminen tarkoittaa pelaajan kannalta pelin voittamista. Pelimaailmoja on tarkoitus ylläpitää useampaa rinnakkain, jotta uudet pelaajat löytäisivät aina maailman, joka on vielä alkuasetelmissa.

3 SUUNNITTELU

3.1 Tutoriaali

Pelin alussa pelaajalle esitellään vanhempi Warcaster-hahmo, joka toimii pelaajan mentorina. Mentori antaa pelaajalle erilaisia tehtäviä, joita suorittamalla pelaaja oppii pelin perusmekaniikat sekä antaa suuntaa strategian luonnille. Alkutehtävät koostuvat lähinnä uusiin rakennuksiin tutustumisesta ja ensimmäisten sotajoukkojen kokoamisesta.

Pelin tutoriaali suunniteltiin tehtävien muotoon, koska konkreettiset tehtävät olivat luonteva ja mielekäs tapa opettaa pelaajille miten peli toimii. Tehtävien lisäksi tutoriaaliin kuuluu käyttöliittymän esittelyruutu, joka esitellään seuraavassa kappaleessa.



KUVIO 1. Mentorin esittely

Opinnäytetyöni aikana kerkesin saada valmiiksi toimivat versiot tehtävistä, jotka opettavat pelaajalle pelin perusmekaniikat. Tämän lisäksi saimme valmiiksi ensimmäiset automaattisesti generoituvat satunnaiset tehtävät, joiden tarkoituksena on luoda lisäsisältöä pelin keskivaiheille. Loppupuolen tehtäviä ei ehditty suunnittelemaan tarkemmin.

Pelaajan opetellessa pelin käyttöliittymän käyttöä, tulee pelin onnistua esittelemään siihen liittyvää ydinmekaniikkaa. Hyvänä käytäntönä tutoriaali-vaiheessa on laittaa pelaaja tekemään niitä asioita, joita hänen tulee osata tehdä jatkossakin. (Järvinen 2010, hakupäivä 5.5.2013.)



KUVIO 2. Ensimmäinen tehtävä

3.2 Graafinen käyttöliittymä

Graafisella käyttöliittymällä (User Interface) tarkoitetaan peleissä niitä peliruudun alueita, joiden kanssa pelaaja voi olla vuorovaikutuksessa tai jotka kertovat pelaajalle jotain oleellista tietoa pelitilanteesta. Tyypillisiä käyttöliittymän osia ovat erilaiset painikkeet ja pisteytykseen liittyvät elementit. Jotta ruutu ei täytyisi käyttöliittymän painikkeista, kaikki edistyneemmät kontrollit laitetaan usein jonkin valikkopainikkeen taakse (Mitchell 2012, 144). Ruudulle jätetään ne painikkeet, joita tarvitaan kaikkein eniten. Suurella kosketusnäytöllä pelatessa on myös järkevää, että painikkeet löytyvät läheltä reunoja ja erityisesti nurkkia, jotta niiden käyttäminen olisi mahdollisimman vaivatonta.

Koska Warcaster on yhtä aikaa selain- ja mobiilipeli, oli graafisen käyttöliittymän suunnittelussa otettava huomioon se, että peliä pelataan sekä hiirellä että kosketusnäytöllä. Tämä tarkoittaa käytännössä sitä, että kaikki pelissä suoritettavat toiminnot tulee onnistua mahdollisimman helposti hiirellä ja kosketuksin. Tärkeää oli siis muistaa testata jokainen uusi asia molemmilla näyttötyypeillä (Dr. Scolastici & Nolte 2013, 163–164). Warcasterissa pelaajan tärkein tehtävä on oman strategian kehittäminen ja toimintojen suorittaminen, joten tärkeintä käyttöliittymälle on sen käytettävyys ja intuitiivisuus, jotta strategian toteuttaminen ja näin ollen pelissä eteneminen tältä kannalta olisi mahdollisimman vaivatonta ja viihdyttävää.

Graafisen käyttöliittymän ulkoasun suunnittelussa pyrittiin ylläpitämään riittävää yksinkertaisuutta. Painikkeiden ja muiden käyttöliittymän elementtien ulkoasu kannattaa pitää myös informatiivisena, jotta niiden merkitys jäisi luontevasti mieleen. (Dr. Scolastici & Nolte 2013, 159.) Esimerkiksi rahaa tarkoittavan pelielementin ulkoasu on hyvä luoda sellaiseksi, että se näyttää rahalta. Silloin sen sisältöä ei tarvitse erikseen selittää. Peliä suunniteltaessa pyrittiin minimoimaan selitettävien asioiden määrää erityisesti siksi, että tutoriaalin sisältöä voidaan tällöin pitää paremmin rajattuna ja tarpeeksi yksinkertaisena.

Ilmanteellisesti pelin kontrollit ovat niin intuitiivisesti toteutettu, että pelaaja todella tuntee olevansa kytkettynä peliin. Kontrollien tulee tuntua luontevilta ja loogisilta. Pelaamisesta saatavan kokemuksen nautinto vähenee, jos jokin pelin elementti erikseen muistuttaa pelaajaa siitä, että kyseessä on pelkkä peli. (Lecky-Thompson 2007, 191.)

Yksi tutoriaaliin kehittämistäni osista oli käyttöliittymän esittely. Tämä ruutu tulee näkyviin, kun pelaaja ensimmäisen kerran kirjautuu peliin sisään, minkä jälkeen ruudun voi avata uudelleen pelin

asetuksista. Esittelyruutua varten minulle annettiin pohjalle konsepti siitä, mitä haetaan (ks. LIITE 1). Käytössäni minulla oli erityisesti esittelyruutua varten tehtyä sekä muualla käyttöliittymässä käytettyä grafiikkaa. Asettelen ja muotoilun määrittelin itse usean kokeilun kautta ja saamaani palautetta hyödyksi käyttäen (ks. KUVIO 3 ja KUVIO 4).



KUVIO 3. Käyttöliittymän esittely, ensimmäinen versio



KUVIO 4. Käyttöliittymän esittely, viimeisin versio

3.3 Käyttäjäkokemus

Graafisen käyttöliittymän suunnittelussa täytyy ottaa huomioon sen luoma käyttäjäkokemus (User Experience), jonka tulisi aina olla käyttäjälle luonnollinen ja helposti omaksuttava. Käyttöliittymää suunniteltaessa pitää aina muistaa, että loppukäyttäjä on ihminen. Käytettävyyden kannalta tärkeintä on pyrkiä siihen, että käyttäjä ymmärtää käyttöliittymän käyttötavat ilman, että sitä täytyy erikseen selittää. Hyvin suunniteltu käyttöliittymä on käyttäjälle ilmiselvä ja helposti käytettävä.

Warcasterin tapauksessa tutoriaali on tärkeä työkalu esittelemään pelaajalle mitä pelissä on mahdollista tehdä ja antaa pelaajalle keskeistä tietotaitoa pelissä pärjäämiseen. Pitkäaikaisena strategiapelinä Warcaster on vaativa peli hallita, mutta helppo omaksua. Tutoriaalin kannalta on siis tärkeää, että pelaaja ymmärtää jo aikaisessa vaiheessa mahdollisimman hyvin kaikki päätöksiin liittyvät vaihtoehdot, mitä hänellä on.

Beta-testauksen aikana Warcasterissa pelaajille annettiin pelin sisäistä rahaa ilmaiseksi. Pelin sisäinen raha on pelin sisäistä valuuttaa, jota voi käyttää mm. nopeuttaakseen pelissä etenemistä. Käytännössä pelin ansaintamalli perustuu suoraan pelin sisäisen valuutan myyntiin niin, että maksavat asiakkaat saavat pienen ylliotteen, mutta ei toisaalta liian suurta ylivoimaa. Vastapainona maksamiselle voi pelaaja investoida enemmän omaa aikaansa pelissä etenemiseen.

Beta-vaiheen ilmaisella valuutalla haluttiin simuloida tilannetta, jossa kaikki pelaajat pystyivät etenemään pelin ihanteellisimmalla vauhdilla. Tällä haluttiin saada mahdollisimman tarkka kuva siitä, kuinka paljon pelin sisäisen valuutan käyttö vaikuttaa pelaajien etenemisnopeuteen. Valuuttaa ei ollut rajoittamattomasti, vaan sitä jaettiin tasaisin aikavälein lisää.

Koska pelin etenemisvauhti oli alusta alkaen suunniteltu verkkaiseksi, oli pelaajien etenemien silmin nähden hidasta ilmaisesta valuutasta huolimatta. Säästeliäänä pelaajana, en oikeastaan osanut käyttää kaikkea sitä valuuttaa, mitä ilmaiseksi sain. Pelaajista pystyi siis erottelemaan melko helposti ne, jotka osasivat kuluttaa resursseja tehokkaasti ja ne, jotka ennemmin säästivät resursseja kaiken varalta myöhemmäksi.

4 TOTEUTTAMINEN

4.1 Unity

Unity on alustariippumaton pelimoottori, jonka käyttötarkoituksena on kehittää videopelejä verkkoliitännäisille (web plugin), työpöytä alustoille, konsoleille ja mobiililaitteille. Opinnäytetyötä aloittaessani Unity-pelimoottori oli yksi parhaista vaihtoehtoista mobiili- ja verkkopelien luomiseen, mutta siitä on sittemmin kehittynyt myös todella laadukas työkalu suuremman luokan konsoli- ja PC-pelien luomiseen.

Pelimoottorin uusinta versiota voi nykyään käyttää täysin ilmaiseksi alusta loppuun pelin menestyksestä riippumatta, mutta maksullisessa versiossa tulee mukana hyödyllisiä työkaluja mm. pelin optimointiin ja analysoimiseen. Aikaisemmassa versiossa ilmaista lisenssiä sai käyttää 3000 dollarin tuloihin asti, jonka jälkeen täytyi maksaa 5 % rojalteja tuloista (Wilde 2015, hakupäivä 14.12.2015). Toisaalta, pelin tulojen tai rahoituksen ylittäessä \$100,000 vuodessa, täytyy Unityn maksullinen versio ottaa käyttöön (Unity Technologies 2015b, hakupäivä 18.12.2015).

Unityn käyttö perustuu sen sisältämän Editorin ja IDE:n yhteistyöhön. Editorin kautta tehdään kaikki muutokset Unityn scene-tiedostoihin, jotka pitävät sisällään tiedon pelin eri "kohtausten" käyttämisestä osasista ja niiden asetuksista. Jakamalla pelin eri kohtauksiin, voi pelin latausaikoja parantaa (Goldstone 2009, 15). IDE:tä käytetään Unity-peliä kehittäessä pääasiassa komentosarjojen eli skriptien kirjoittamiseen.

4.1.1 Editori

Editorissa on käytössä kaksi näkymää: Scene-näkymä ja Game-näkymä. Scene-näkymän puolella tehdään kaikki näkyvät muutokset editorin graafista käyttöliittymää hyödyntäen, ja sen kautta voidaan vapaasti tarkastella editorin generoimaa kolmiulotteista ympäristöä eri kuvakulmista. Game-näkymä sen sijaan näyttää aina sen, mitä pelaajan näkökulmasta ruudulla tapahtuu.

Graafista käyttöliittymää toteuttaessa käytettiin omia kameraolioita käyttöliittymälle ja pelaajan omalle perspektiiville. Käyttöliittymän kameran sijainti Editorin kolmiulotteisessa tilassa on irrelevantti, sillä se on asetettu näyttämään pelaajan ruudulle vain käyttöliittymän komponentit. Kamera-oliot näyttävät yhdistettynä pelaajan oman perspektiivin sekä sen eteen piirtyvät käyttöliittymän komponentit.



KUVIO 5. Scene-näkymä

Scene-näkymässä voidaan tarkastella ja hallita yksittäisiä olioita tai olioryhmiä. KUVIO 5:ssa näkyy käyttöliittymään aseteltuja olioita ja niiden ääriviivoja. Oliot sisältävät sekä visuaalista että toiminnollista sisältöä, kuten napin toiminnollisuutta.



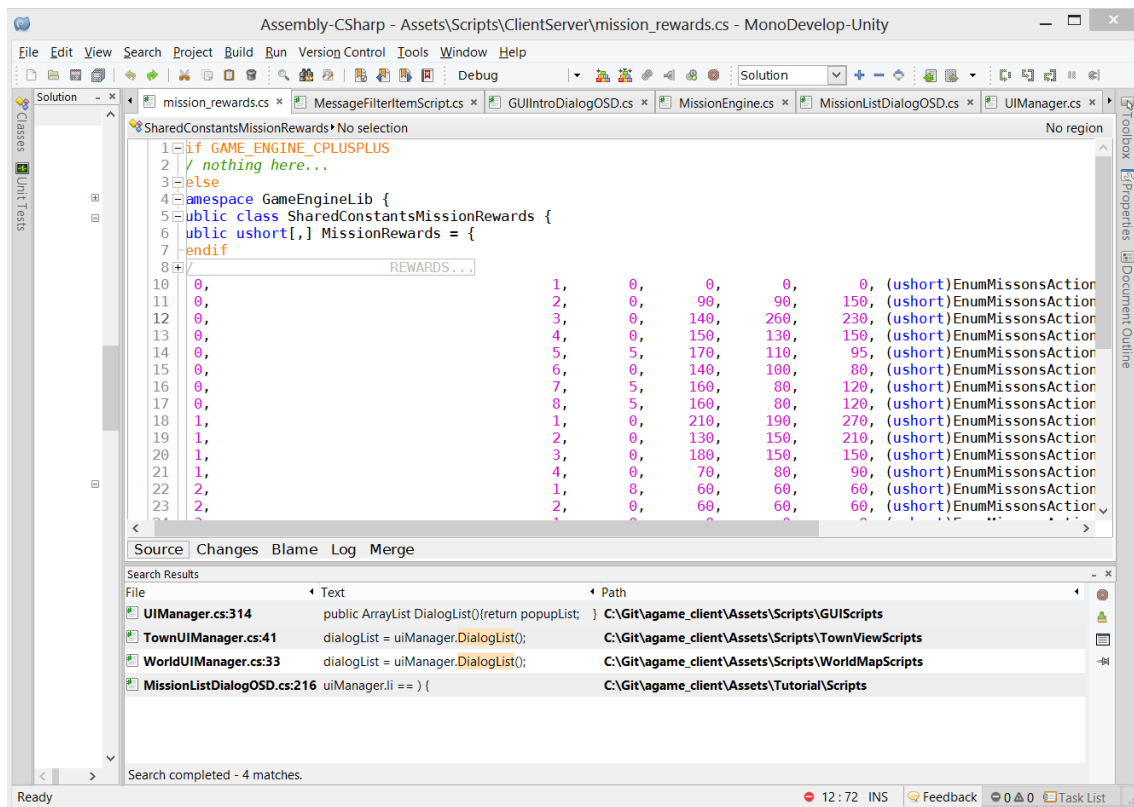
KUVIO 6. Game-näkymä

Game-näkymässä ilmenee se, mitä pelaajan ruudulle piirtyy. KUVIO 6 sisältää myös muita Editorin osia, kuten avattuna olevan scene-tiedoston hierarkianäkymä, valitun olion tarkemmat tiedot (inspector) sekä koko projektin sisältämät tiedostopolut (Goldstone 2009, 17).

4.1.2 MonoDevelop IDE

IDE (integrated development environment) on monipuolinen virheitä jäljittävä ja korjaava ohjelmointiympäristö, joka sisältää lähdekoodin kirjoittamiseen tarvittavat työkalut. Unityn IDE:nä toimii MonoDevelop, joka perustuu alustariippumattomaan vapaan lähdekoodin kehitysympäristöön Monoon (Wikipedia 2015, hakupäivä 14.12.2015).

Pelien kehittäjillä on MonoDevelop IDE:n tukemana käytettävissään kolme ohjelmointikieltä: C# (lausunta: C-sharp), UnityScript ja Boo, joista UnityScriptiä kutsutaan tutummalla yleisnimellä JavaScriptiksi, johon se perustuu. Koska itse pelimoottorin kehitysympäristö perustuu suoraan .NET Frameworkiin, samalla tavalla kuin C#, voidaan C#:a pitää tehokkaimpana kielenä käytettäväksi tällä peli-moottorilla. Opinnäytetyön pelissä käytettiin Unity-pelimoottorin puolella C#-ohjelmointikieltä.



KUVIO 7. MonoDevelop IDE

MonoDevelop IDE:n vaihtoehtona toimii nykyään myös Visual Studio, joka on C#-ohjelmointia varten alusta alkaen kehitetty IDE. Tuki Visual Studiota varten tuli vasta myöhemmässä vaiheessa, mutta se pitää sisällään paljon hyödyllisiä ja edistyneempiä työkaluja ohjelmointityötä varten, kunhan valittuna ohjelmointikielenä on C#. (Unity Technologies 2015a, hakupäivä 16.12.2015.)

4.2 Git

Git-versionhallintajärjestelmä on ohjelmoinnin kannalta tärkein työkalu, johon tutustuin opinnäyte-työni aikana. Git on hajautettu versionhallintatyökalu, jonka avulla ohjelmointiprojekteissa voidaan varmistaa, että kaikki hajautetusti suoritettu kehitystyö pystytään yhdistämään ja että kaikkia hyväksytyjä muutoksia pystytään tarkastelemaan myös jälkeenpäin kronologisessa järjestyksessä.

Kaikki muutokset, jotka ohjelmointiprojektiin on tehty, tallentuvat erillisiin committeihin. Commitit ovat eriteltyjä koodimuutoksia, joita voi valmistella useamman kappaleen ennen projektiin yhdistämistä. Commitit voi joko suoraan yhdistää projektin versionhallintaan tai niistä voidaan luoda pull request, joka vaatii erillisen hyväksynnän ennen yhdistämistä. Käytännössä on suositeltavaa sisällyttää yksi ydinajatus yhteen committiin, jotta versiohistoria säilyisi helposti luettavana. Hyväksytyt muutokset tulevat voimaan projektin ajankohtaisimmaksi versioksi, jota vasten jatkossa lisätyt commitit joudutaan yhteensovittamaan.

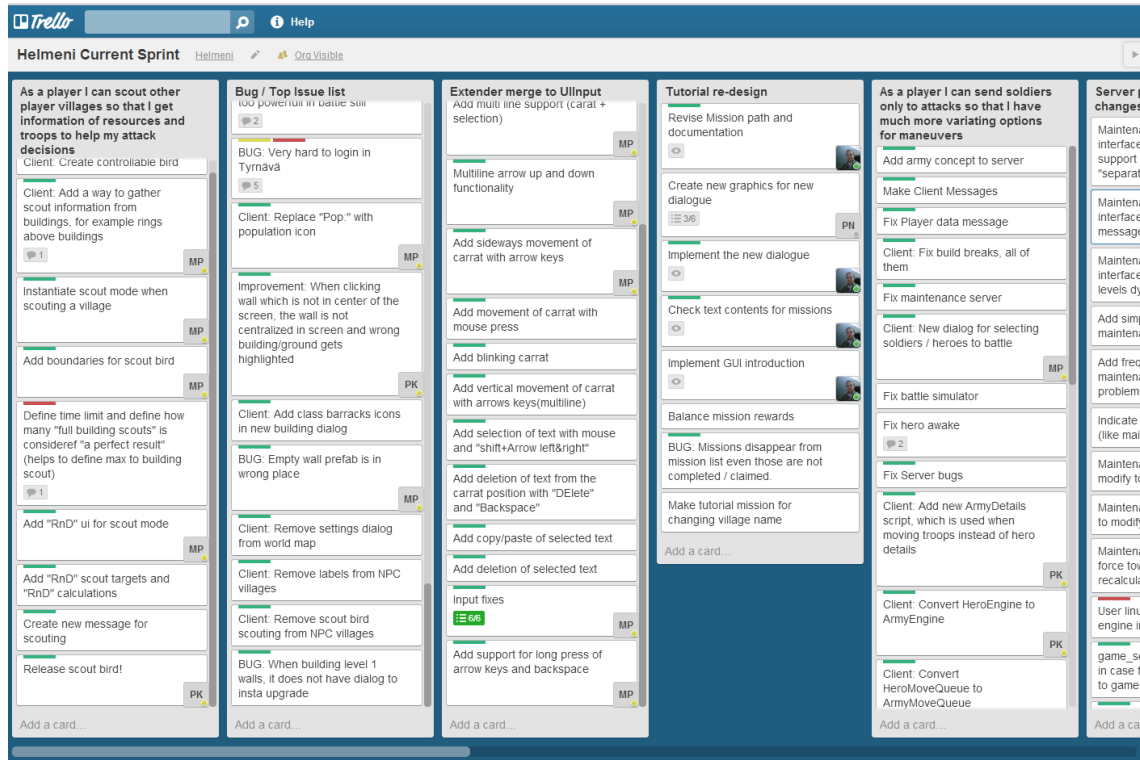
Git-versionhallinnalle on olemassa useita verkkopalveluita kuten Github tai Bitbucket, jotka sisältävät Git-versionhallinnan lisäksi muita hyödyllisiä työkaluja kuten esimerkiksi haku, jonka avulla on helppo etsiä koodinpätkiä hakusanojen avulla. Jos käyttäjä haluaa löytää tietyn funktion tai tiedoston sijainnin projektissa, onnistuu se kätevästi käyttämällä hakua suoraan kyseisen funktion tai tiedoston omalla nimellä. Muita yleisiä työkaluja ovat bug-trakkeri, kehitystoiveet, tehtävien hallinta, wiki sekä esimerkiksi projektin eri versioiden lataamisen mahdollistava julkaisujärjestelmä.

4.3 Ketterät menetelmät

Käyttämämme kehittämismenetelmät perustuivat Scrum-viitekehykseen ja nk. ”kanban-taulun” käyttöön, joskin tyypillistä vapaammassa muodossa. Scrum-viitekehystä seuraten listasimme tehtäviä kehitysjonoon ja valikoimme tietyn määrän tehtäviä muutaman viikon pituisiin sprintteihin. Itse kehitystyön visualisointi suoritettiin ”kanban-taulua” käyttämällä.

Ketterää ohjelmistokehitystä käyttämällä pyrimme pilkkomaan isommat kehitystehtävät pienemmiksi, jotta niiden edistymistä pystyi seuraamaan ja tilastoimaan paremmin. Pieneksi pureksittu

tehtävä oli myös yleensä helpompi toteuttaa lyhyessä ajassa, mikä ylläpiti tehokasta työtahtia. Pieneksi pilkottuja tehtäviä hallitsimme verkkopalvelu Trellon avulla. Trello on verkkopohjainen projektihallintatyökalu, joka perustuu ”kanban-taulun” käyttöön.



KUVIO 8. Trello

Trellossa työtehtävät visualisoitiin taululle post-it muistilapuiksi, joita pystyi muokkaamaan ja liikuttelemaan pitkin taulua, kun tehtävä eteni. Projektiä varten tehtiin useampi taulu joita myös käytettiin hieman eri tavoin. Esimerkiksi meneillään olevaa kehitystyön sprinttiä varten listattiin muistilaput riippuen siitä, mihin se liittyi. Yhden isomman tehtävän alle listattiin useampi pienempi lappu, joiden etenemistä merkattiin väreillä.

Työpäivät aloitimme kokoontumalla yhteiseen stand-up tapaamiseen, jonka aikana jokainen työntekijä kertoi yksi kerrallaan aikaisemman työpäivän etenemisestä sekä alkavan päivän tavoitteista. Samalla kerroimme mahdollisista haasteista tai tarpeista etenemiseen liittyen. Stand-up tapaamisen tarkoituksena oli ylläpitää tietoa siitä, miten kaikkien työtehtävät etenevät ja tuoda mahdollisia ongelmia muiden tietoon jo aikaisessa vaiheessa. Tapaamiset pidettiin nimensä mukaisesti tyypilliseen tapaan seisten, jotta ne pysyivät tarpeeksi lyhyinä ja mahdollisimman vähän aikaa vievinä. Työviikkojen alussa sama tapaaminen toistettiin, mutta työpäivän sijaan kerrottiin koko työviikon etenemisestä yleisemmällä tasolla.

Koska kehitystiimi oli pieni ja kehitystyö haluttiin pitää mahdollisimman nopeana, emme käyttäneet erikseen aikaa koodin laadun tarkasteluun. Ohjelmoinnin yhtenä suurena haasteena onkin yhtenäisen ja toimivan syntaksin ylläpitäminen samalla kun koodia kirjoitetaan. Ohjelmoinnissa saman toiminnollisuuden voi lähes aina toteuttaa hyvinkin monella eri tavalla, mikä voi aiheuttaa paljon ylimääräistä työtä silloin, kun toimitaan yhteistyössä useamman ohjelmoijan kanssa. Yleinen tapa pitää koodin kirjoitusasu yhtenäisenä on määritellä etukäteen, millaista kirjoitusasun standardia tulee käyttää. Tärkeintä on joka tapauksessa se, että koodi toimii tehokkaasti ja on helposti ymmärrettävää. (McSchaffry 2012, 99–103.)

5 GAMESCOM

Helmenin johtoporras oli suunnitellut matkan Gamescom-tapahtumaan Saksaan, mutta yksi lähti-
jöistä ei lopulta pystynytkään osallistumaan. Lopputuloksena pääsin siis osallistumaan hänen si-
jastaan matkaan, jonka yhtenä tärkeänä päämääränä oli luoda pelille tunnettavuutta ja löytää kiin-
nostuneista kävijöistä beta-testaajia pelillemme.

Matka oli äärimmäisen hyödyllinen kokemus, sillä tapahtuman aikana pyrimme esittelemään peliä
mahdollisimman monelle tapahtuman osallistujalle. Omalta kannaltani hyödyllisintä oli päästä esit-
telemään pelin pitchiä, eli myyntipuhetta. Meillä oli esittelypöytä ja muutama testilaite sekä iso tv,
jossa pyöri näyttävä traileri pelistä. Usean päivän aikana kerkesimme pitchaamaan peliä sadoille
kävijöille. Oulu Game Labissa suorittamani harjoittelun aikana yksi tärkeimmistä opeista oli nimen-
maan pitchin eli myyntipuheen esittäminen, jota harjoiteltiin useaan otteeseen.

Gamescomin esittelypöydällä erikoisinta oli se, että suurimman kilpailijan, Travianin, oma pöytä oli
aivan vastapäätä meistä. Kävimme jututtamassa Travianin edustajaa ja saimme häneltä pari oheis-
tuotettakin kotiin vietäväksi.



KUVIO 9. Gamescom esittelypöytä

6 POHDINTA

Pelikehitys on laaja ala ja sitä ei opi muuten kuin tekemällä. Opinnäytetyötä aloittaessani oli minulla aiempaa kokemusta pelikehityksestä aikaisemmalta puolelta vuodelta, jolloin suoritin opintojeni harjoittelun Oulu Game Labissa. Ilman vastaavaa pelikehitystaustaa olisi aihe ollut liian laaja toteuttavaksi.

Työn aihe oli haasteita täynnä ja opittavaa oli paljon edessä. Esimerkiksi grafiikan ja fonttien käsittely Unityn Editorissa vaativat paljon aikaa, jotta ne saatiin optimoitua näkymään pelissä halutulla tavalla. Lisäksi tutoriaalin tehtävälogiikan rakentaminen alusta alkaen oli iso prosessi, mutta yhteistyössä muiden ohjelmoijien kanssa siitä saatiin toimiva kokonaisuus.

Tärkeitä oppeja työtä tehdessä tuli sekä työkalujen käytöstä että ketterien työprosessien omaksumisesta. Työn arvioitiin kestävän n. kolme kuukautta, mutta työsuhdetta jatkettiin vielä tämän jälkeen parilla kuukaudella kesän ajaksi omalla työsopimuksella. Tämän ansiosta pystyin jatkamaan opinnäytetyöni aiheen parissa myös pidemmälle.

LÄHTEET

Goldstone, W. 2009. Unity Game Development Essentials. Birmingham: Packt Publishing Ltd.

Järvinen, A. 2010. First Five Minutes: How Tutorials Make or Break Your Social Game. Hakupäivä 5.5.2013, http://www.gamasutra.com/view/feature/132715/first_five_minutes_how_tutorials_.php.

Lecky-Thompson, G. 2007. Video Game Design Revealed. Boston: Course Technology / Cengage Learning.

Mitchell, B. 2012. Game Design Essentials. Indianapolis: John Wiley & Sons, Inc.

Dr. Scolastici, C & Nolte, D. 2013. Mobile Game Design Essentials. Birmingham: Packt Publishing Ltd.

Unity Technologies. 2015a. Visual Studio C# Integration. Hakupäivä 16.12.2015, <http://docs.unity3d.com/Manual/VisualStudioIntegration.html>.

Unity Technologies. 2015b. FAQ: LICENSING & ACTIVATION. Hakupäivä 18.12.2015, <https://unity3d.com/unity/faq>.

Wikipedia. 2015. MonoDevelop. Viitattu 14.12.2015, <https://en.wikipedia.org/wiki/MonoDevelop>.

Wilde, Tyler. 2015. Unity 5 out now, Personal Version free with no royalties. Hakupäivä 14.12.2015, <http://www.pcgamer.com/unity-5-out-now-personal-version-free-with-no-royalties>.

At the beginning of game, the PC has been brought to his village and the thing he needs to concentrate on at start is building. The mentor warcaster, who works as mentor and advisor, first explains about basic interactions with GUI to the player and lastly tells about his list of village matters that need to be worked out (= task list).

Missions/Tasks will be advancing in their own routes. First route's first task is marked as 1.1, First route's second task 1.2 and so on. One task per route is active at a time. The tasks act as a tool to teach player how to do things and what is expected of the player.

GUI Introduction

- Indicate to the player all objects that he/she can interact with.
- Opens up automatically on first login. Can be opened again from the top element dialog
- Technically, place a popup dialog, with only one button for closing it (avoid misclicks). Dialog has Mentor Warcaster explaining about all elements on the GUI (what can be clicked, what is what.). The elements should also have some sort of textblocks under/over the elements with short title name for the specific element. The Mentor Warcaster should end the speech with telling about his **task list**.

*Character to be used in the tutorial missions, The PC's mentor: Mentor WarCaster Alemethush

Placeholder and first version of GUI Introduction dialogue:

"Welcome, young WarCaster! It's time to introduce you to your graphical user interface, so pay attention! You are able to interact with all the stationary elements on your screen, so be sure to try all of them out. They should be quite self-explanatory in the way they work, but more information can be found from the wiki.

Can you see the glowing button in the bottom left corner? Press that to find out more about your pressing duties. To make things simple, you are shown only the most important ones at a time. Still, your aim should always be to grow your influence and this usually involves making sure that your opponents **can't** do so as well.

Come on, we haven't got all day!"

You can open this introduction dialog later on from the element found on top of your screen."

0. Route available at the start

0.1: Change village name

0.2: Build Builder's house, to gain access to building slots

reward: resources and small amount of premium currency

0.3: Build Farmhouse, so village gains food production

reward: some resources

0.4: Build Sawmill, Stonemason's hut and Smelter, so village gains resource income

reward: some resources

0.5: Build Warehouse, so village has more space for resources

reward: some resources

0.6: Build Garrison

reward: gold and soldiers

0.7: Build Swords Arena

reward: 5? soldiers

0.8: Build Tavern and hire a hero

reward: gold

0.9: Recruit x-amount of soldiers

1. Route becomes active after *task 0.5* is done

1.1: Build 4 Farms

reward: some resources

1.2: Upgrade wood, stone and iron production to 20/h

reward: some resources

1.3: Upgrade Warehouse to level 2

reward: some resources

2. Route becomes active after *task 0.5* is done

2.1: Build a stash

reward: some resources

2.2.: Build wall

3. Route becomes active after *task 0.8* is done

3.2: Raid a village